JSON-LD 1.1
An onramp* to the Semantic Web

Harold Solbrig and Dazhi Jiao
Johns Hopkins University
March 10, 2020

*** and *offramp***

OUTLINE

- Short review of RDF and JSON
- Getting from JSON to RDF
- @context - the secret sauce
- Summary

Note

- A lot of this presentation has hyperlinks, often in the form of a TinyURL. If you've got the slides:

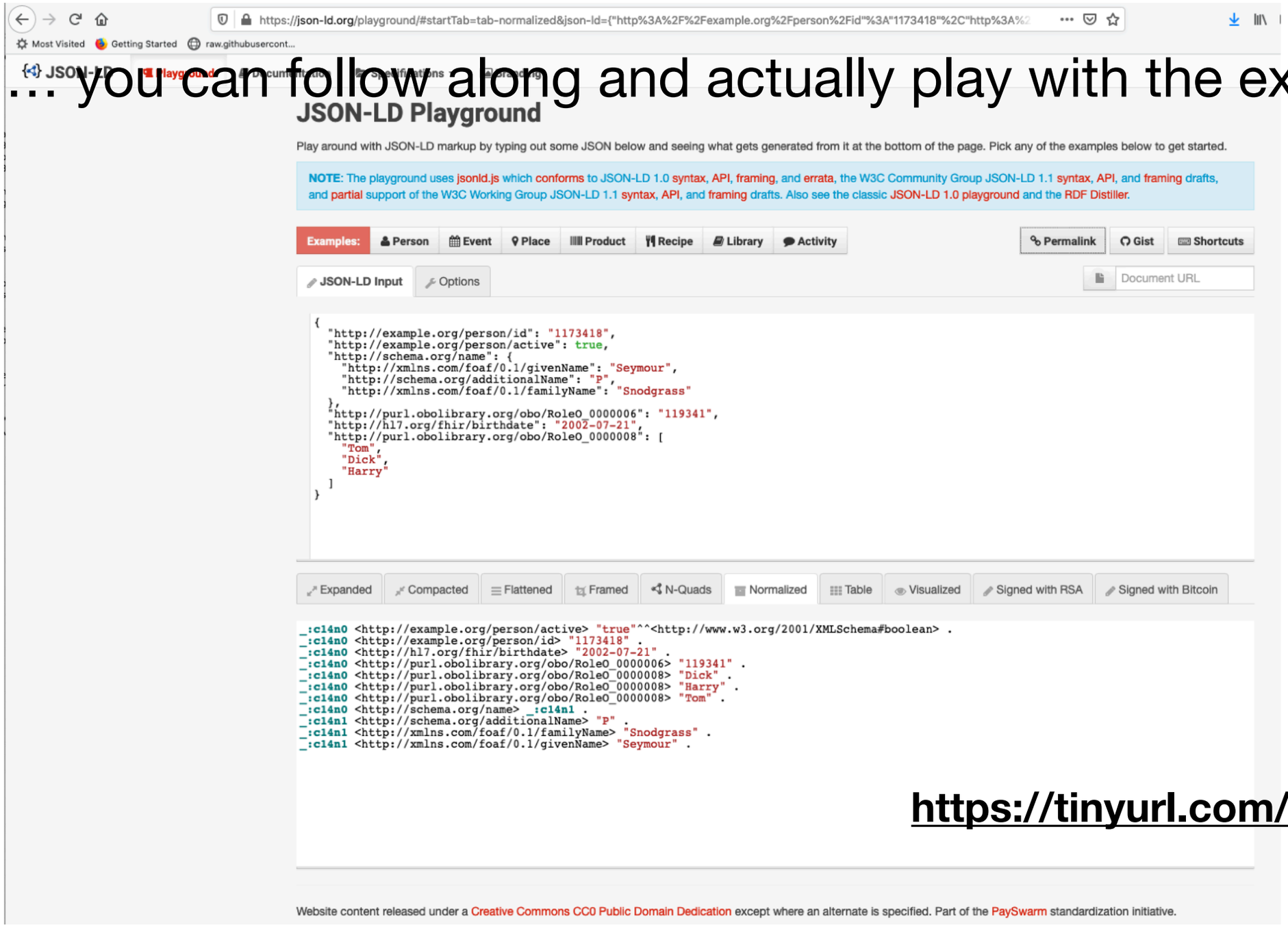
https://github.com/fhircat/fhir_rdf_validator/tree/master/tutorial/us2ts/slides.pdf

or

https://tinyurl.com/s8ux8tn …

Notes

… you can follow along and actually play with the examples



https://tinyurl.com/vb9dq5v

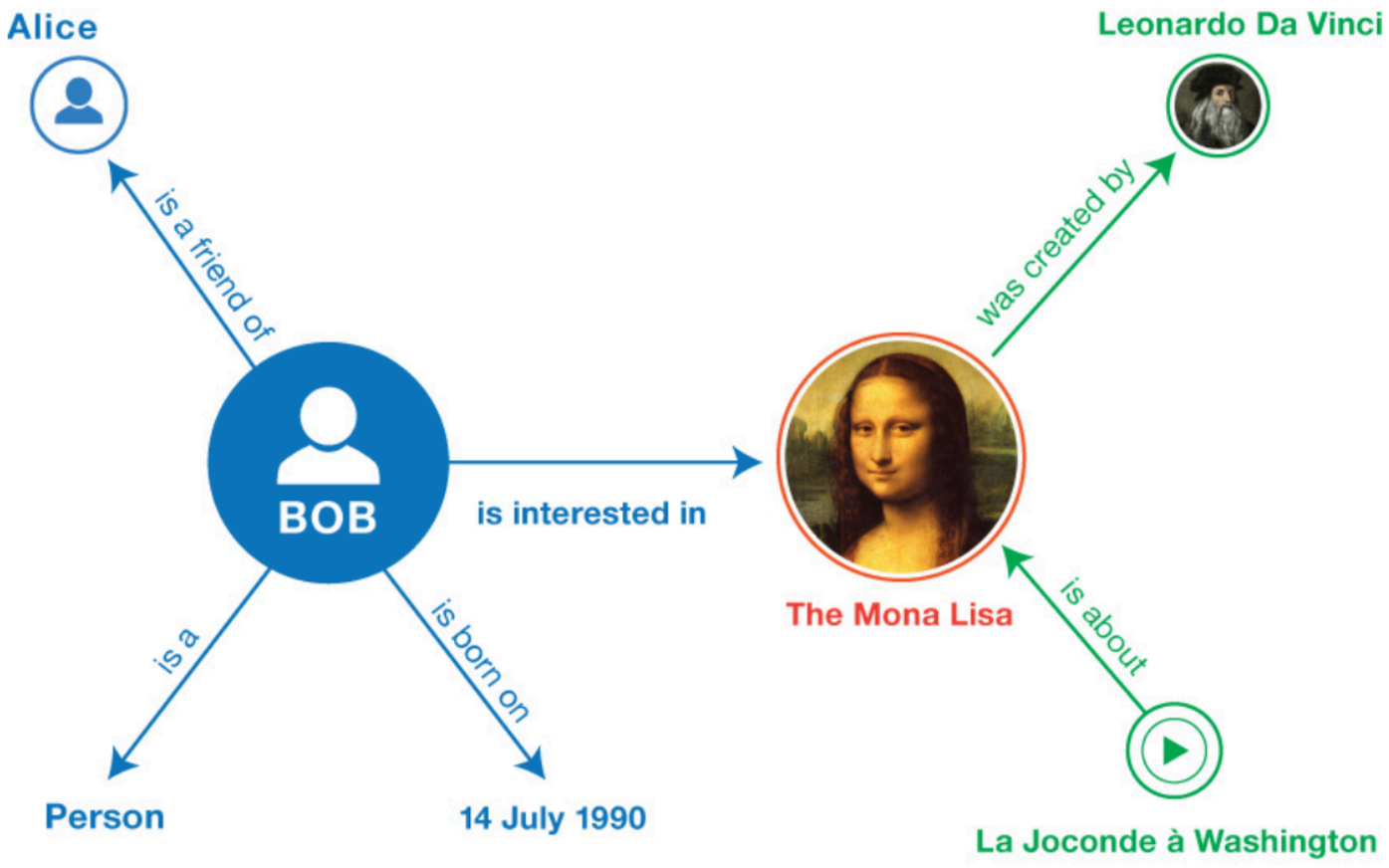
Short Review of RDF

The Resource Description Format (RDF)

The RDF 1.1 Primer has an excellent introduction. Key points:

- **RDF allows us to make *statements* about *resources.***
- **A resource can be anything**
- **RDF *statements* take the form:**
  <subject> <predicate> <object>
  where *subject* and *object* represent resources and *predicate* represents the nature of their relationship
- **An RDF *graph* is a (possibly named) collection of statements.**

RDF Data Model




```
<Bob> <is a> <person>.
<Bob> <is a friend of> <Alice>.
<Bob> <is born on> <the 4th of July 1990>.
<Bob> <is interested in> <the Mona Lisa>.
<the Mona Lisa> <was created by> <Leonardo da Vinci>.
<the video 'La Joconde à Washington'> <is about> <the Mona Lisa>
```

RDF Triples

```
<subject> <predicate> <object>
```

RDF is an abstraction of a *directed graph (digraph)*, where *subject* and *object* represent vertices and *predicate* represents an edge.

- <subject> - is either an IRI or a Blank node (BNode)
- <predicate> - IRI
- <object> - is either an IRI, BNode or Literal

Key point for later: a digraph can contain *cycles* and does necessarily have a single root (i.e. is not necessarily a *tree*)

IRI’s

We (more or less) know what these look like …

https://en.wikipedia.org/

https://ieeexplore.ieee.org/document/771073

https://www.w3.org/TR/rdf11-primer/#section-IRI

https://schema.org/Person

The key feature being that an IRI can *reference* (link to) information that is *shared* and *open.* The RDF data model and URI’s form the backbone of an architecture and philosophy known as Linked (Open) Data (LD).

Literals

- An RDF object can be a value — a string, a date, a number, etc.

  “La Joconde”, “the 4th of July, 1990”, “3.14159”

- A literal can be associated with a *datatype,* which “enables such values to be parsed and interpreted correctly”

- A literal datatype offers no guarantee of conformance. *“fred”^^xsd:integer* is acceptable RDF

```
“La Joconde”   ≡   “La Joconde”^^xsd:string
“the 4th of July, 1990”^^xsd:date
“3.14159”^^xsd:double
“(17, 2, France)”^^ex:myspecialType
```

Literals

- String literals can be associated with a language tag

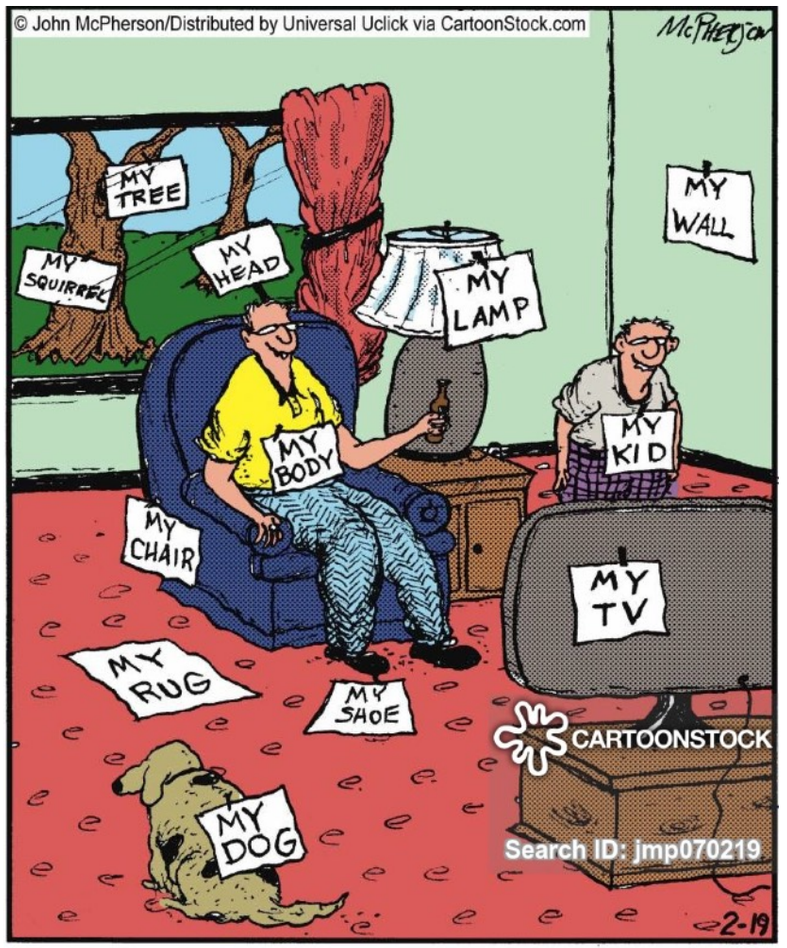
```
“bacteria”@en
“bactéries”@fr
“bacterium”@de
```

```
“analog”@en-us
“analogue”@en-gb
```

Blank Nodes (BNodes)

Use 1: To avoid having to assign a URI to *everything*

"Now! *That* should clear up a few things around here!"



The president of MySpace.com at home.

Blank Nodes
Use 1

Avoid having to assign a URI to *everything*

The president of MySpace.com at home.

:Harold ex:owns :HaroldsRug, :HaroldsWall,
:HaroldShoe, :HaroldsDog, …

:HaroldsRug a ont:Carpet ;
ex:colored :Red ;
ex:quality :Ugly .

:Harold ex:owns _:bn1 .
_:bn1 a ont:Carpet ;
ex:colored :Red ;
ex:quality :Ugly .

***Not* an identifier.**
Valid only in the context of the current graph

Blank Nodes
Use 1

To avoid having to assign a URI to *everything*

The president of MySpace.com at home.

```
:Harold ex:owns _:bn1 .
_:bn1 a ont:Carpet ;
      ex:colored :Red ;
      ex:quality :Ugly .

:Harold ex:owns _:a12345 .
_:a12345 a ont:Carpet ;
      ex:colored :Red ;
      ex:quality :Ugly .

:Harold ex:owns [
      a ont:Carpet ;
      ex:colored :Red ;
      ex:quality :Ugly ] .
```

All three of these represent identical graphs (!)

Turtle syntax uses “[“ to avoid assigning an identifier at all…

Blank Nodes Use 2

Determiner phrases

We don't know its name… its identity *is* its description (from our perspective).

```
_:b17245 a <http://schema.org/house> ;
        ex:laterality ont:Left ;
        ex:location "Taylorsvile, FL" ;
        ex:street "Hillside Lane" .
```

Side note: The is an existential assertion. The graph above satisfied* iff there is an interpretation that makes this true. It is NOT a definition (all instances of :_b17245 are also instances of houses that…) !!!

*** As defined in https://www.w3.org/TR/2014/REC-rdf11-mt-20140225/#rdf-interpretations**

RDF Serialization Formats

RDF Triples (NTriples)

```
<http://example.org/Sam> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
<http://example.org/Sam> <http://example.org/name> _:b1 .
_:b1 <http://xmlns.com/foaf/0.1/givenName> "Sam" .
_:b1 <http://xmlns.com/foaf/0.1/familyName> "Smith" .
<http://example.org/Sam> <http://xmlns.com/foaf/0.1/knows> _:b2 .
_:b2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> <http://example.org/Melissa> .
_:b2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:b3 .
_:b3 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> <http://example.org/Dazhi> .
_:b3 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> <http://www.w3.org/1999/02/22-rdf-syntax-ns#nil> .
<http://example.org/Melissa> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
<http://example.org/Dazhi> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
```

Sam (as defined in http://example.org) is a Person (as defined by FOAF)
Sam's first name is "Sam" (using FOAF's definition of "givenName")
Sam's last name is "Smith" (using FOAF's definition of "familyName")
Sam knows (FOAF's definition of knows) Melissa
Sam knows Dazhi

RDF Serialization Formats

RDF Triples (NTriples)

```
<http://example.org/Sam> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
<http://example.org/Sam> <http://example.org/name> _:b1 .
_:b1 <http://xmlns.com/foaf/0.1/givenName> "Sam" .
_:b1 <http://xmlns.com/foaf/0.1/familyName> "Smith" .
<http://example.org/Sam> <http://xmlns.com/foaf/0.1/knows> _:b2 .
_:b2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> <http://example.org/Melissa> .
_:b2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:b3 .
_:b3 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> <http://example.org/Dazhi> .
_:b3 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> <http://www.w3.org/1999/02/22-rdf-syntax-ns#nil> .
<http://example.org/Melissa> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
<http://example.org/Dazhi> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
```

Equivalent to:

RDF Turtle

```
@prefix : <http://example.org/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

:Sam a foaf:Person ;
    :name [ foaf:familyName "Smith" ;
            foaf:givenName "Sam" ] ;
    foaf:knows ( :Melissa :Dazhi ) .

:Melissa a foaf:Person .
 :Dazhi a foaf:Person .
```

RDF Serialization Formats

RDF Triples (NTriples)

```
<http://example.org/Sam> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
<http://example.org/Sam> <http://example.org/name> _:b1 .
_:b1 <http://xmlns.com/foaf/0.1/givenName> "Sam" .
_:b1 <http://xmlns.com/foaf/0.1/familyName> "Smith" .
<http://example.org/Sam> <http://xmlns.com/foaf/0.1/knows> _:b2 .
_:b2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> <http://example.org/Melissa> .
_:b2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:b3
_:b3 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> <htt
_:b3 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> <htt
<http://example.org/Melissa> <http://www.w3.org/1999/02/22
<http://example.org/Dazhi> <http://www.w3.org/1999/02/22-r
```

RDF XML

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:foaf="http://xmlns.com/foaf/0.1/"
 xmlns="http://example.org/"
>
<foaf:Person rdf:about="http://example.org/Sam">
   <name>
    <rdf:Description rdf:nodeID="ub2bL8C11">
     <foaf:givenName>Sam</foaf:givenName>
     <foaf:familyName>Smith</foaf:familyName>
    </rdf:Description>
   </name>
   <foaf:knows rdf:parseType="Collection">
    <rdf:Description rdf:about="http://example.org/Melissa"/>
    <rdf:Description rdf:about="http://example.org/Dazhi"/>
   </foaf:knows>
 </foaf:Person>
 <foaf:Person rdf:about="http://example.org/Dazhi"/>
 <foaf:Person rdf:about="http://example.org/Melissa"/>
</rdf:RDF>
```

Equivalent to:

RDF Turtle

```
@prefix : <http://example.org/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

:Sam a foaf:Person ;
   :name [ foaf:familyName "Smith" ;
           foaf:givenName "Sam" ] ;
   foaf:knows ( :Melissa :Dazhi ) .

:Melissa a foaf:Person .
:Dazhi a foaf:Person .
```

RDF Serialization Formats

RDF Turtle

```
@prefix : <http://example.org/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

:Sam a foaf:Person ;
    :name [ foaf:familyName "Smith" ;
            foaf:givenName "Sam" ] ;
    foaf:knows ( :Melissa :Dazhi ) .

:Melissa a foaf:Person .
 :Dazhi a foaf:Person .
```

XML

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns="http://example.org/"
>
<foaf:Person rdf:about="http://example.org/Sam">
    <name>
      <rdf:Description rdf:nodeID="ub2bL8C11">
        <foaf:givenName>Sam</foaf:givenName>
        <foaf:familyName>Smith</foaf:familyName>
          </rdf:Description>
```

Different representations of *exactly* the same information

RDF Triples (NTriples)

```
<http://example.org/Sam> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
<http://example.org/Sam> <http://example.org/name> _:b1 .
_:b1 <http://xmlns.com/foaf/0.1/givenName> "Sam" .
_:b1 <http://xmlns.com/foaf/0.1/familyName> "Smith" .
<http://example.org/Sam> <http://xmlns.com/foaf/0.1/knows> _:b2 .
_:b2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> <http://example.org/Melissa> .
_:b2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:b3 .
_:b3 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> <http://example.org/Dazhi> .
_:b3 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> <http://www.w3.org/1999/02/22-rdf-syntax-ns#nil> .
<http://example.org/Melissa> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
 <http://example.org/Dazhi> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
```

JSON-LD is* just another format (!)

Turtle

```
:Sam a foaf:Person;
    :name [foaf:givenName "Sam";
           foaf:familyName "Smith"];
    foaf:knows (:Melissa :Dazhi) .
:Melissa a foaf:Person .
```

XML

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF>
  <foaf:Person rdf:about="http://example.org/Sam">
    <foaf:knows rdf:parseType="Collection">
      <rdf:Description rdf:about="http://example.org/Melissa"/>
      <rdf:Description rdf:about="http://example.org/Dazhi"/>
    </foaf:knows>
      <name>
```

Different representations of *exactly* the same information

JSON-LD

```
[
 {
  "@id": "http://example.org/Sam",
  "@type": [ "http://xmlns.com/foaf/0.1/Person" ],
   "http://example.org/name": [
     {  "@id": "_:ub3bL8C11" }
  ],
  {   "@id": "_:ub3bL8C11",
      "http://xmlns.com/foaf/0.1/familyName": [
       {  "@value": "Smith"    }
      ],
      "http://xmlns.com/foaf/0.1/givenName": [
       {  "@value": "Sam"    }
      ]
    },
  "http://xmlns.com/foaf/0.1/knows": [
   {  "@list": [
         {  "@id": "http://example.org/Melissa"},
```

*** JSON-LD is actually a whole lot more, but still, at its core, this is what JSON-LD is really all about.**

JSON-LD
Just another RDF Serialization Format…?

JSON-LD
Just another RDF Serialization Format…?

Even Shorter Review of JSON

JavaScript Object Notation (JSON)

See: https://www.json.org/json-en.html for everything you need to know.

Components:

- *object* — a (possibly empty) collection of *name/value* pairs
- *array* — an (possibly empty) ordered list of *values*
- *value* — one of: *string, number, boolean, null, object or array*

A JSON document is either an *array* or an *object.*

JSON Example

```
{   name      value
  “name": “BigCocoInc",
  "type": "Company",
  "people": {
     "Sam": {
       "name": {
          "first": “Sam",
          “middle”: null,
          "last": “Smith”,
          “age”: 47
        },
       "employees": ["Melissa", “Dazhi"]
    },
    "Melissa": {
      “living”: true
      "name": {
        "last": "Johnson"
      },
    }
}
```

list

objects

JSON
Key Points

- JSON objects represents *trees*
- JSON supports a small, fixed set of data types:
  - String, number, boolean, null, object, list
- Object keys must be strings
- A key cannot appear more than once in an object
- Lists are *ordered* and duplicates are allowed
- There is no notion of object identifiers (subjects), so there is no notion of *reference* (within the JSON model…)

Getting from JSON to RDF

JSON to RDF

Start with arbitrary JSON structure:


```
{
  "id": "1173418",
  "active": true,
  "name": {
    "first": "Seymour",
    "middle": "P",
    "last": "Snodgrass"
  },
  "father": "119341",
  "birthdate": "2002-07-21",
  "children": [
    "Tom",
    "Dick",
    "Harry"
  ]
}
```


https://tinyurl.com/vvqd8yg

JSON to RDF

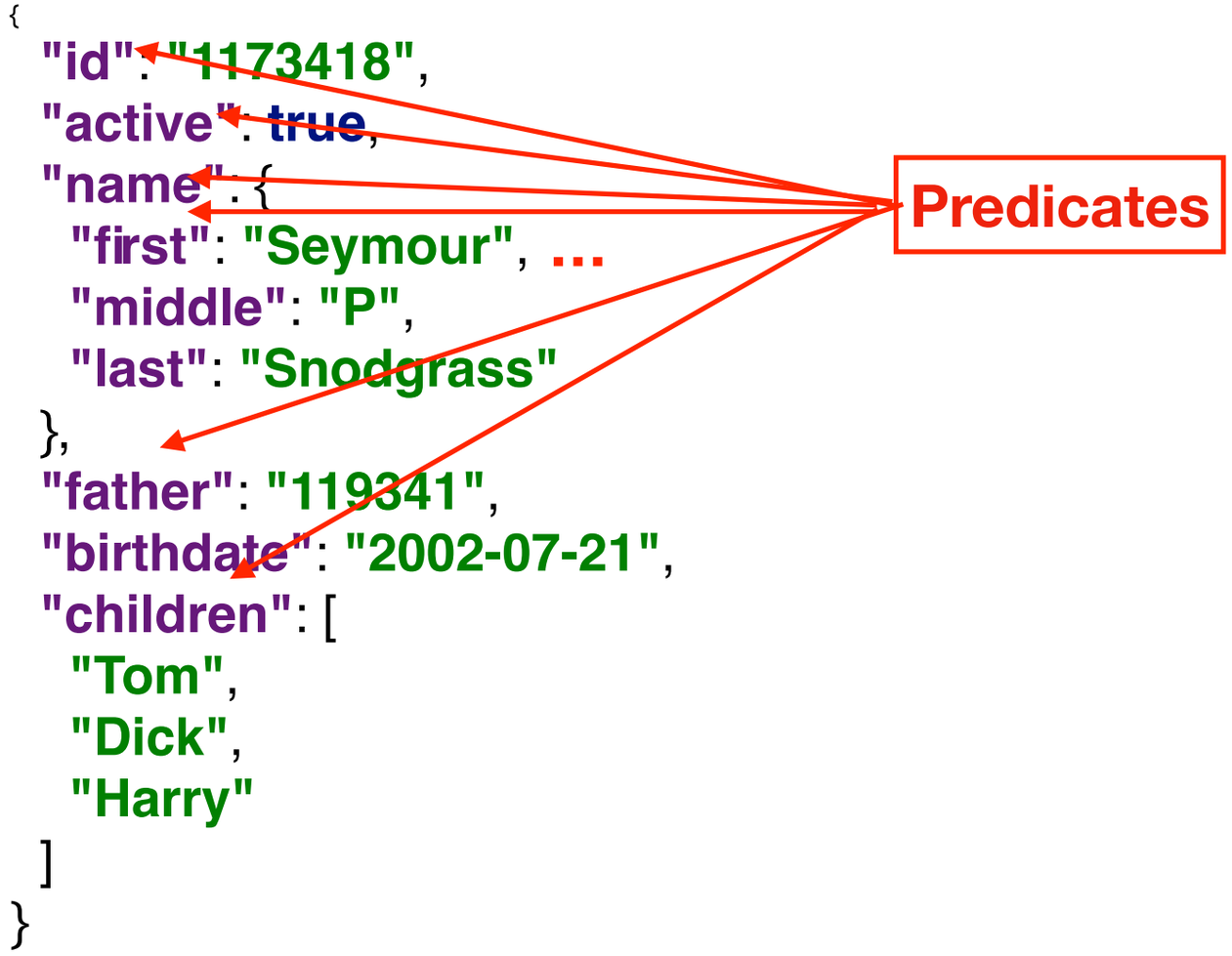
Start with arbitrary JSON structure:

```
{
  "id": "1173418",
  "active": true,
  "name": {
    "first": "Seymour",
    "middle": "P",
    "last": "Snodgrass"
  },
  "father": "119341",
  "birthdate": "2002-07-21",
  "children": [
    "Tom",
    "Dick",
    "Harry"
  ]
}
```

https://tinyurl.com/vvqd8yg

JSON to RDF

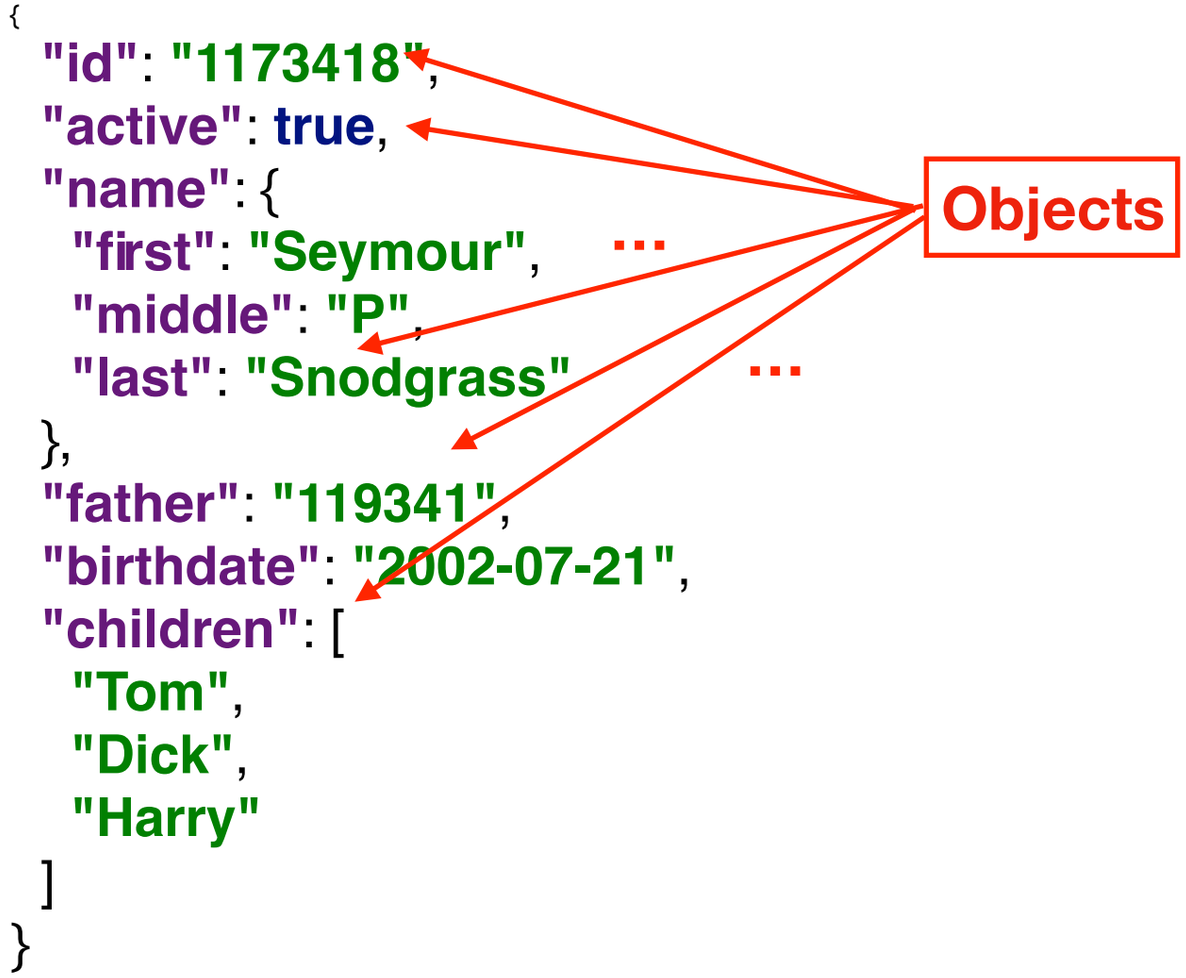
Start with arbitrary JSON structure:



https://tinyurl.com/vvqd8yg

JSON to RDF

Start with arbitrary JSON structure:



https://tinyurl.com/vvqd8yg

JSON to RDF Conversion Steps

1. **Convert predicates to URI's** - identify or, as a last resort, create the URI that represents the semantics of the corresponding JSON key

Convert Predicates to URI's
This is the hard (and *key*) step

This is the core of Linked Open Data — what do I (we) mean when I (we) say:

- **"id"**
- **"active"**
- **"name"**
- **"name.first"**
- **"name.middle"**
- **"name.last"**
- **"birthdate"**
- **"children"**

in *this* particular context?

Predicates to URIs
We'll start with a default …

{ "**http://example.org/person/**id": "1173418",
  "**http://example.org/person/**active" : true,
  "**http://example.org/person/**name" : {
      "**http://example.org/person/**first" : "Seymour",
      "**http://example.org/person/**middle" : "P",
      "**http://example.org/person/**last": "Snodgrass"
   },
   "**http://example.org/person/**father": "119341",
   "**http://example.org/person/**birthdate" : "2002-07-21",
   "**http://example.org/person/**children" : ["Tom", "Dick", "Harry"]
}

https://tinyurl.com/vb9dq5v

Poof! We've got RDF

JSON-LD Playground

Play around with JSON-LD markup by typing out some JSON below and seeing what gets generated from it at the bottom of

NOTE: The playground uses jsonld.js which conforms to JSON-LD 1.0 syntax, API, framing, and errata, the W3C Communi
and partial support of the W3C Working Group JSON-LD 1.1 syntax, API, and framing drafts. Also see the classic JSON-LD

Examples: Person | Event | Place | Product | Recipe | Library | Activity

JSON-LD Input | Options

```
{
  "http://example.org/person/id": "1173418",
  "http://example.org/person/active": true,
  "http://example.org/person/name": {
    "http://example.org/person/first": "Seymour",
    "http://example.org/person/middle": "P",
    "http://example.org/person/last": "Snodgrass"
  },
  "http://example.org/person/father": "119341",
  "http://example.org/person/birthdate": "2002-07-21",
  "http://example.org/person/children": [
    "Tom",
    "Dick",
    "Harry"
  ]
}
```

Expanded | Compacted | Flattened | Framed | N-Quads | Normalized | Table | Visua

```
_:b0 <http://example.org/person/active> "true"^^<http://www.w3.org/2001/XMLSchema#boolean> .
_:b0 <http://example.org/person/birthdate> "2002-07-21" .
_:b0 <http://example.org/person/children> "Dick" .
_:b0 <http://example.org/person/children> "Harry" .
_:b0 <http://example.org/person/children> "Tom" .
_:b0 <http://example.org/person/father> "119341" .
_:b0 <http://example.org/person/id> "1173418" .
_:b0 <http://example.org/person/name> _:b1 .
_:b1 <http://example.org/person/first> "Seymour" .
_:b1 <http://example.org/person/last> "Snodgrass" .
_:b1 <http://example.org/person/middle> "P" .
```

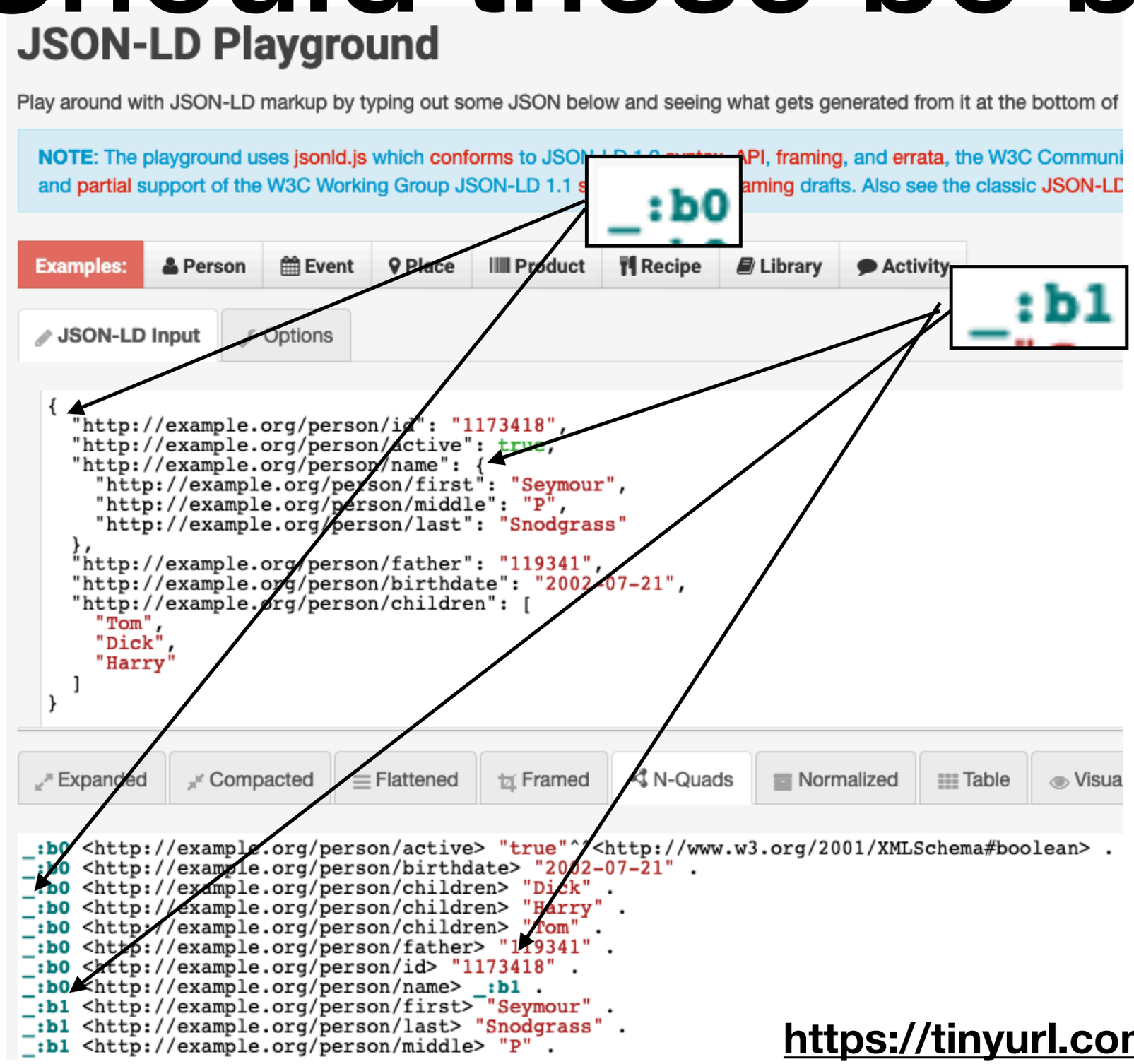
https://tinyurl.com/vb9dq5v

Predicates to URIs (The hard part)

{ “**http://example.org/person/**id”: “1173418”,
“**http://example.org/person/**active” : true,
“**http://schema.org/name**” : {
“**http://xmlns.com/foaf/0.1/givenName**” : “Seymour”,
“**http://schema.org/additionalName**” : “P”,
“**http://xmlns.com/foaf/0.1/familyName**”: “Snodgrass”
},
“**http://purl.obolibrary.org/obo/RoleO_0000006**": “119341”,
“**http://hl7.org/fhir/**birthdate” : “2002-07-21”,
“**http://purl.obolibrary.org/obo/RoleO_0000008**” : [“Tom”, “Dick”, “Harry”]
}

https://tinyurl.com/usxdezn

Should these be blank?



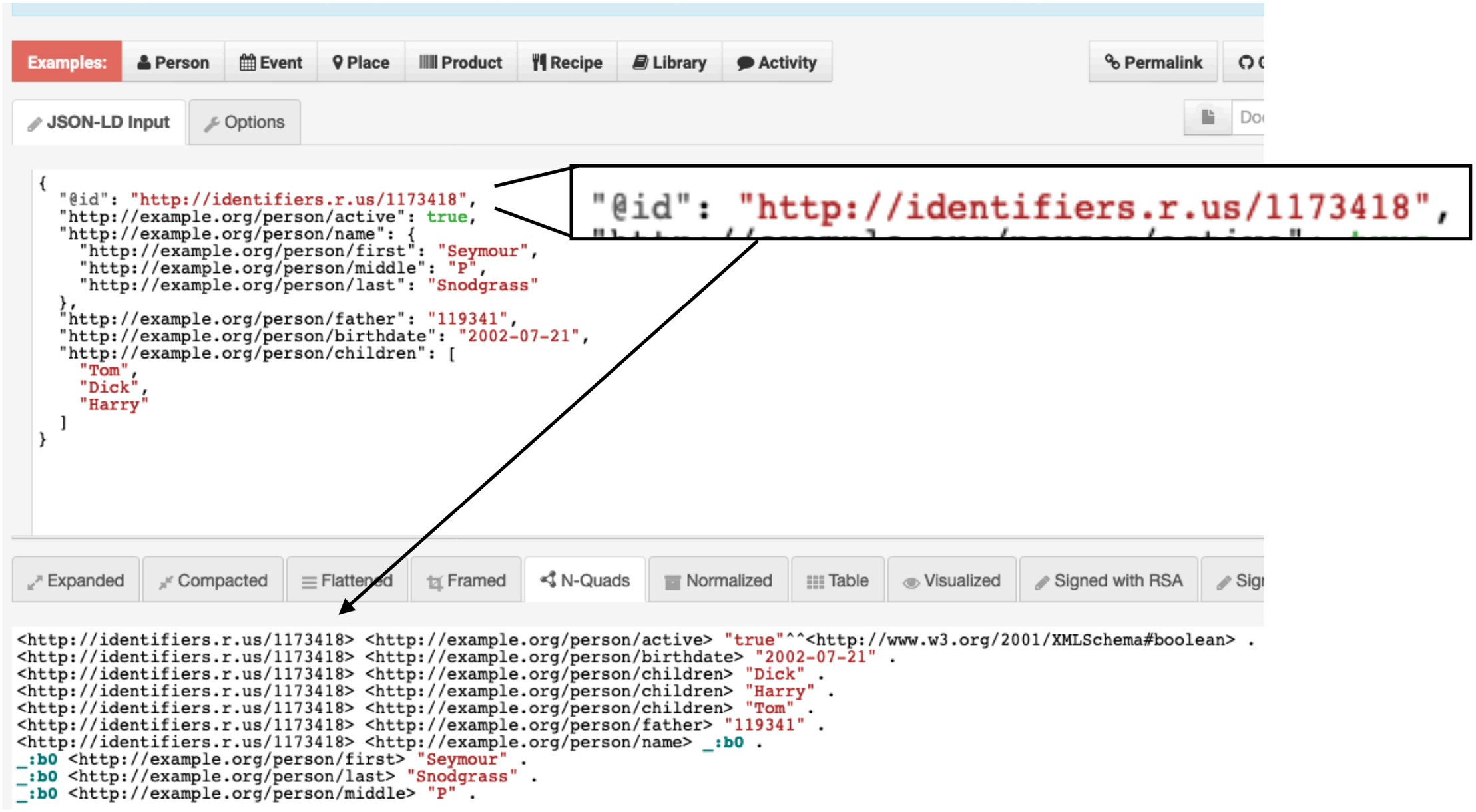
https://tinyurl.com/vb9dq5v

JSON to RDF Conversion Steps

1. Convert predicates to URI’s

2. Add subject URI’s where appropriate

Change “id” to a subject



https://tinyurl.com/sodryej

Could do inner subjects…

```
{
  "@id": "http://identifiers.r.us/1173418",
  "http://example.org/person/active": true,
  "http://example.org/person/name": {
    "@id": "http://id.r.us/SeymouresName",
    "http://example.org/person/first": "Seymour",
    "http://example.org/person/middle": "P",
    "http://example.org/person/last": "Snodgrass"
  },
  "http://example.org/person/father": "119341",
  "http://example.org/person/birthdate": "2002-07-21",
  "http://example.org/person/children": [
    "Tom",
    "Dick",
    "Harry"
  ]
}
```

"@id": "http://id.r.us/SeymouresName",

Expanded | Compacted | Flattened | Framed | N-Quads | Normalized | Table | Visualized | Signed with RSA

```
<http://id.r.us/SeymouresName> <http://example.org/person/first> "Seymour" .
<http://id.r.us/SeymouresName> <http://example.org/person/last> "Snodgrass" .
<http://id.r.us/SeymouresName> <http://example.org/person/middle> "P" .
<http://identifiers.r.us/1173418> <http://example.org/person/active> "true"^^<http://www.w3.org/2001/XMLSchema#boolean> .
<http://identifiers.r.us/1173418> <http://example.org/person/birthdate> "2002-07-21" .
<http://identifiers.r.us/1173418> <http://example.org/person/children> "Dick" .
<http://identifiers.r.us/1173418> <http://example.org/person/children> "Harry" .
<http://identifiers.r.us/1173418> <http://example.org/person/children> "Tom" .
<http://identifiers.r.us/1173418> <http://example.org/person/father> "119341" .
<http://identifiers.r.us/1173418> <http://example.org/person/name> <http://id.r.us/SeymouresName> .
```

https://tinyurl.com/v7q8b38

Data Types

Examples: | Person | Event | Place | Product | Recipe | Library | Activity | Permalink | Gis

JSON-LD Input | Options | Docu

Built-in JSON type

```
{
  "@id": "http://identifiers.r.us/1173418",
  "http://example.org/person/active": true,
  "http://example.org/person/name": {
    "http://example.org/person/first": "Seymour",
    "http://example.org/person/middle": "P",
    "http://example.org/person/last": "Snodgrass"
  },
  "http://example.org/person/father": "119341",
  "http://example.org/person/birthdate": "2002-07-21",
  "http://example.org/person/children": [
    "Tom",
    "Dick",
    "Harry"
  ]
}
```

Expanded | Compacted | Flattened | Framed | N-Quads | Normalized | Table | Visualized | Signed with RSA | Signe

```
<http://identifiers.r.us/1173418> <http://example.org/person/active> "true"^^<http://www.w3.org/2001/XMLSchema#boolean> .
<http://identifiers.r.us/1173418> <http://example.org/person/birthdate> "2002-07-21" .
<http://identifiers.r.us/1173418> <http://example.org/person/children> "Dick" .
<http://identifiers.r.us/1173418> <http://example.org/person/children> "Harry" .
<http://identifiers.r.us/1173418> <http://example.org/person/children> "Tom" .
<http://identifiers.r.us/1173418> <http://example.org/person/father> "119341" .
<http://identifiers.r.us/1173418> <http://example.org/person/name> _:b0 .
_:b0 <http://example.org/person/first> "Seymour" .
_:b0 <http://example.org/person/last> "Snodgrass" .
_:b0 <http://example.org/person/middle> "P" .
```

https://tinyurl.com/sodryej

JSON to RDF Conversion Steps

1. Convert predicates to URI's
2. Add subject URI's where appropriate
3. Add explicit datatypes and language tags

Builtin JSON Data Types

JSON-LD Input | Options

```
{
  "@id": "http://ex.org/builtins",
  "http://ex.org/builtins/str": "abc",
  "http://ex.org/builtins/strreal": 17.2,
  "http://ex.org/builtins/strurl": "http://google.org/",
  "http://ex.org/builtins/strdate": "2012-01-17",
  "http://ex.org/builtins/strint": 1,
  "http://ex.org/builtins/strnegint": -17,
  "http://ex.org/builtins/strdbl": 245.999631999631,
  "http://ex.org/builtins/strbool": false,
  "http://ex.org/builtins/strnull": null
}
```

Expanded | Compacted | Flattened | Framed | N-Quads | Normalized | Table | Visualized | Signed with RSA | Signed

```
<http://ex.org/builtins> <http://ex.org/builtins/str> "abc" .
<http://ex.org/builtins> <http://ex.org/builtins/strbool> "false"^^<http://www.w3.org/2001/XMLSchema#boolean> .
<http://ex.org/builtins> <http://ex.org/builtins/strdate> "2012-01-17" .
<http://ex.org/builtins> <http://ex.org/builtins/strdbl> "2.45999631999631E2"^^<http://www.w3.org/2001/XMLSchema#double> .
<http://ex.org/builtins> <http://ex.org/builtins/strint> "1"^^<http://www.w3.org/2001/XMLSchema#integer> .
<http://ex.org/builtins> <http://ex.org/builtins/strnegint> "-17"^^<http://www.w3.org/2001/XMLSchema#integer> .
<http://ex.org/builtins> <http://ex.org/builtins/strreal> "1.72E1"^^<http://www.w3.org/2001/XMLSchema#double> .
<http://ex.org/builtins> <http://ex.org/builtins/strurl> "http://google.org/" .
```

https://tinyurl.com/qwbmqmj

Assigning Data types

JSON-LD Input | Options | Document

```
{
  "@id": "http://ex.org/builtins",
  "http://ex.org/builtins/str": "abc",
  "http://ex.org/builtins/strreal": {
    "@value": 17.2,
    "@type": "http://www.w3.org/2001/XMLSchema#decimal"
  },
  "http://ex.org/builtins/strurl": {"@value": "http://google.org/", "@type": "http://www.w3.org/2001/XMLSchema#anyURI"},
  "http://ex.org/builtins/strdate": {"@value": "2012-01-17","@type": "http://www.w3.org/2001/XMLSchema#date"},
  "http://ex.org/builtins/strint": 1,
  "http://ex.org/builtins/strnegint": -17,
  "http://ex.org/builtins/strdbl": {"@value": 245.999631999631, "@type": "http://www.w3.org/2001/XMLSchema#double"},
  "http://ex.org/builtins/strbool": false,
  "http://ex.org/builtins/strnull": null
}
```

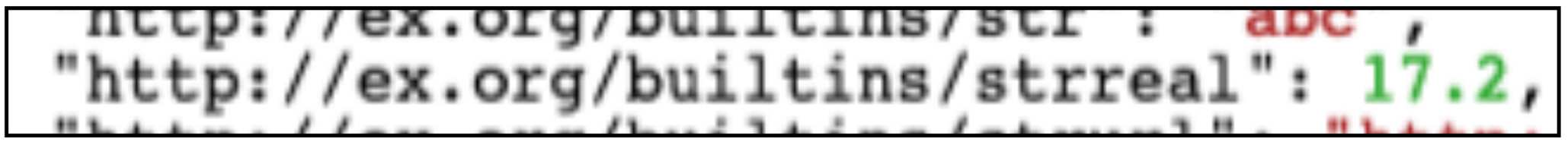
```
"http://ex.org/builtins/strreal": {
  "@value": 17.2,
  "@type": "http://www.w3.org/2001/XMLSchema#decimal"
},
```

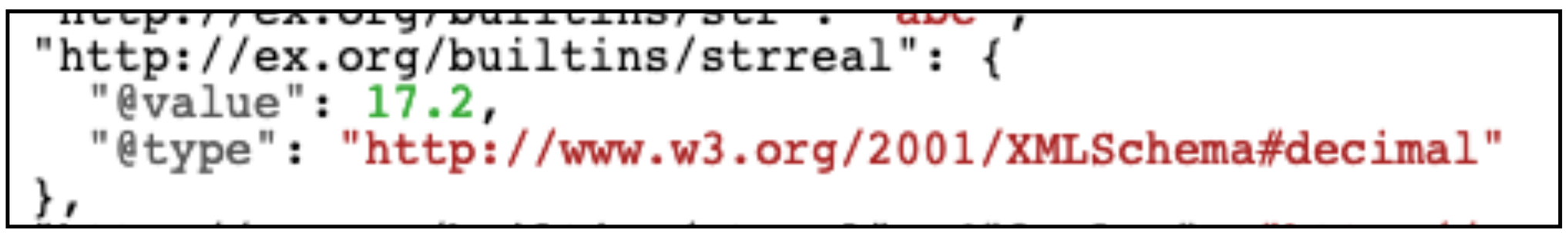
Expanded | Compacted | Flattened | Framed | N-Quads | Normalized | Table | Visualized | Signed with RSA | Signed with

```
<http://ex.org/builtins> <http://ex.org/builtins/str> "abc" .
<http://ex.org/builtins> <http://ex.org/builtins/strbool> "false"^^<http://www.w3.org/2001/XMLSchema#boolean> .
<http://ex.org/builtins> <http://ex.org/builtins/strdate> "2012-01-17"^^<http://www.w3.org/2001/XMLSchema#date> .
<http://ex.org/builtins> <http://ex.org/builtins/strdbl> "2.45999631999631E2"^^<http://www.w3.org/2001/XMLSchema#double> .
<http://ex.org/builtins> <http://ex.org/builtins/strint> "1"^^<http://www.w3.org/2001/XMLSchema#integer> .
<http://ex.org/builtins> <http://ex.org/builtins/strnegint> "-17"^^<http://www.w3.org/2001/XMLSchema#integer> .
<http://ex.org/builtins> <http://ex.org/builtins/strreal> "1.72E1"^^<http://www.w3.org/2001/XMLSchema#decimal> .
<http://ex.org/builtins> <http://ex.org/builtins/strurl> "http://google.org/"^^<http://www.w3.org/2001/XMLSchema#anyURI> .
```

https://tinyurl.com/wrwyj5a

What just happened?





The native JSON doesn’t allow datatype assignment.

To do something like assign a type, we have to create an “out of band” object to start to add things.

All Equivalent in JSON-LD (but *not* JSON …)

```
"http://ex.org/builtins/strreal": 17.2,
```

Value

```
"http://ex.org/builtins/strreal": {
  "@value": 17.2
},
```

Object w/ a single “@value” element

```
"http://ex.org/builtins/strreal": [{
  "@value": 17.2
}],
```

List *containing* a single Object w/ a single “@value” element

JSON-LD treats all three of these as representing the *same* thing…

- **Example 1 is called “Compact” syntax**
- **Example 3 is called “Expanded” syntax**

The Expanded syntax allows us to add “@type”’s (and a lot more)

Changing data types

JSON-LD Input | Options | Document URL

```
{
  "@id": "http://identifiers.r.us/1173418",
  "http://example.org/person/active": true,
  "http://example.org/person/name": {
    "http://example.org/person/first": "Seymour",
    "http://example.org/person/middle": "P",
    "http://example.org/person/last": "Snodgrass"
  },
  "http://example.org/person/father": "119341",
  "http://example.org/person/birthdate": {
    "@value": "2002-07-21",
    "@type": "http://www.w3.org/2001/XMLSchema#date"},
  "http://example.org/person/children": [
    "Tom",
    "Dick",
    "Harry"
  ]
}
```

```
"http://example.org/person/birthdate": {
  "@value": "2002-07-21",
  "@type": "http://www.w3.org/2001/XMLSchema#date"},
```

Expanded | Compacted | Flattened | Framed | N-Quads | Normalized | Table | Visualized | Signed with RSA | Signed with Bit

```
<http://identifiers.r.us/1173418> <http://example.org/person/active> "true"^^<http://www.w3.org/2001/XMLSchema#boolean> .
<http://identifiers.r.us/1173418> <http://example.org/person/birthdate> "2002-07-21"^^<http://www.w3.org/2001/XMLSchema#date> .
<http://identifiers.r.us/1173418> <http://example.org/person/children> "Dick" .
<http://identifiers.r.us/1173418> <http://example.org/person/children> "Harry" .
<http://identifiers.r.us/1173418> <http://example.org/person/children> "Tom" .
<http://identifiers.r.us/1173418> <http://example.org/person/father> "119341" .
<http://identifiers.r.us/1173418> <http://example.org/person/name> _:b0 .
_:b0 <http://example.org/person/first> "Seymour" .
_:b0 <http://example.org/person/last> "Snodgrass" .
_:b0 <http://example.org/person/middle> "P" .
```

https://tinyurl.com/wveqpyt

Language strings and JSON

https://tinyurl.com/tknnohm

- No standard way to do it
- All sorts of options

https://tinyurl.com/s49dxfp

```
{
  "id": "14189004",
  "fsn_en": "Measles (disorder)",
  "pref_en": "Measles",
  "alt_en_1": "Morbilli",
  "alt_en_2": "Rubeola",
  "fsn_fr_ca": "rugeole (trouble)",
  "alt_fr_ca": "rugeole",
  "pref_da": "Mæslinger"
}
```

```
{
  "id": "14189004",
  "descriptions": {
    "fsn": {
       "en": "Measles (disorder)",
       "fr-ca": "rugeole (trouble)"
    },
    "pref": {
       "en": "Measles",
      " da": "Mæslinge"
    },
    "alt": {
      "en": [
          "Morbilli",
          "Rubeola"
        ],
      "fr-ca": [
        "rugeole"]
    }
  }
}
```

https://tinyurl.com/wown6bu

```
{
  "id": "14189004",
  "fsn": [
    {
      "value": "Measles (disorder)",
      "language": "en"
    },
    {
      "value": "rugeole (trouble)",
      "language": "fr_ca"
    }
  ],
  "pref": [
    {
      "value": "Measles",
      "language": "en"
    },
    {
      "value": "rugeole",
      "language": "fr_ca"
    },
    {
      "value": "Mæslinger",
      "language": "da"
    }
  ],
  "alt": [
    {
      "value": "Morbilli",
      "language": "en"
    },
    {
      "value": "Rubeola",
      "language": "en"
    },
    {
      "value": "rugeole",
      "language": "ca_fr"
    }
    ]
}
```

Language strings and JSON-LD

JSON-LD Input | Options

```
{
  "@id": "http://snomed.info/id/14189004",
  "http://snomed.info/id/900000000000003001": [
    {
      "@value": "Measles (disorder)",
      "@language": "en"
    },
     {
      "@value": "rugeole (trouble",
      "@language": "fr-ca"
    }
  ],
  "http://www.w3.org/2004/02/skos/core#prefLabel": [
    {
      "@value": "Measles",
      "@language": "en"
    },
    {
      "@value": "Mæslinger",
      "@language": "da"
    }
  ],
  "http://www.w3.org/2004/02/skos/core#altLabel": [
     {
      "@value": "Morbilli",
      "@language": "en"
    },
    {
      "@value": "Rubeola",
      "@language": "en"
    },
    {
      "@value": "rugeole",
      "@language": "fr-ca"
    }
  ]
}
```

Our goal

We can *edit* any of the preceding examples to get here, but the transformations are non-trivial.

We will learn how to use JSON-LD contexts to perform these edits for us.

Expanded | Compacted | Flattened | Framed | N-Quads | Normalized | Table | Visualized | Signe

```
<http://snomed.info/id/14189004> <http://snomed.info/id/900000000000003001> "Measles (disorder)"@en .
<http://snomed.info/id/14189004> <http://snomed.info/id/900000000000003001> "rugeole (trouble"@fr-ca .
<http://snomed.info/id/14189004> <http://www.w3.org/2004/02/skos/core#altLabel> "Morbilli"@en .
<http://snomed.info/id/14189004> <http://www.w3.org/2004/02/skos/core#altLabel> "Rubeola"@en .
<http://snomed.info/id/14189004> <http://www.w3.org/2004/02/skos/core#altLabel> "rugeole"@fr-ca .
<http://snomed.info/id/14189004> <http://www.w3.org/2004/02/skos/core#prefLabel> "Measles"@en .
<http://snomed.info/id/14189004> <http://www.w3.org/2004/02/skos/core#prefLabel> "Mæslinger"@da .
```

https://tinyurl.com/to3e7c5

List Ordering

JSON-LD Input | Options

```
{
  "@id": "http://identifiers.r.us/1173418",
  "http://example.org/person/active": true,
  "http://example.org/person/name": {
    "http://example.org/person/first": "Seymour",
    "http://example.org/person/middle": "P",
    "http://example.org/person/last": "Snodgrass"
  },
  "http://example.org/person/father": "119341",
  "http://example.org/person/birthdate": {
    "@value": "2002-07-21",
    "@type": "http://www.w3.org/2001/XMLSchema#date"
  },
  "http://example.org/person/children": [
    "Tom",
    "Dick",
    "Harry"
  ]
}
```

List order *not* preserved (!)

Expanded | Compacted | Flattened | Framed | N-Quads | Normalized | Table

```
<http://identifiers.r.us/1173418> <http://example.org/person/active> "true"^^<http://www
<http://identifiers.r.us/1173418> <http://example.org/person/birthdate> "2002-07-21"^^<h
<http://identifiers.r.us/1173418> <http://example.org/person/children> "Dick" .
<http://identifiers.r.us/1173418> <http://example.org/person/children> "Harry" .
<http://identifiers.r.us/1173418> <http://example.org/person/children> "Tom" .
<http://identifiers.r.us/1173418> <http://example.org/person/father> "119341" .
<http://identifiers.r.us/1173418> <http://example.org/person/name> _:c14n0 .
_:c14n0 <http://example.org/person/first> "Seymour" .
_:c14n0 <http://example.org/person/last> "Snodgrass" .
_:c14n0 <http://example.org/person/middle> "P" .
```

https://tinyurl.com/twzj3xp

JSON to RDF Conversion Steps

1. Convert predicates to URI's
2. Add subject URI's where appropriate
3. Add data types, language tags and/or direction
4. Add list ordering

List Ordering

```
"http://example.org/person/children": {
"@list": [
    "Tom",
    "Dick",
    "Harry"
  ]
```

```
{
  "@id": "http://identifiers.r.us
  "http://example.org/person/acti
  "http://example.org/person/name
    "http://example.org/person/fi
    "http://example.org/person/mi
    "http://example.org/person/la
  },
  "http://example.org/person/fath
  "http://example.org/person/birt
    "@value": "2002-07-21",
    "@type": "http://www.w3.org/2001/XMLSchema#date"
  },
  "http://example.org/person/children": {
    "@list" : [
      "Tom",
      "Dick",
      "Harry"
    ]
  }
}
```

Expanded | Compacted | Flattened | Framed | N-Quads | Normalized | Table

```
<http://identifiers.r.us/1173418> <http://example.org/person/active> "true"^^<http://
<http://identifiers.r.us/1173418> <http://example.org/person/birthdate> "2002-07-21"^
<http://identifiers.r.us/1173418> <http://example.org/person/children> _:c14n0 .
<http://identifiers.r.us/1173418> <http://example.org/person/father> "119341" .
<http://identifiers.r.us/1173418> <http://example.org/person/name> _:c14n2 .
_:c14n0 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> "Tom" .
_:c14n0 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:c14n1 .
_:c14n1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> "Dick" .
_:c14n1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:c14n3 .
_:c14n2 <http://example.org/person/first> "Seymour" .
_:c14n2 <http://example.org/person/last> "Snodgrass" .
_:c14n2 <http://example.org/person/middle> "P" .
_:c14n3 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> "Harry" .
_:c14n3 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> <http://www.w3.org/1999/02/
```

List Ordering (turtle)

```
"http://example.org/person/children": {
"@list": [
    "Tom",
    "Dick",
    "Harry"
  ]
```

```
@prefix ns1: <http://example.org/person/> .
@prefix ns2: <http://hl7.org/fhir/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://identifiers.r.us/1173418> ns1:active true ;
    ns1:children ( "Tom" "Dick" "Harry" ) ;
    ns1:father "http://identifiers.r.us/119341" ;
    ns2:Person.name [ ns1:first "Seymour" ;
            ns1:last "Snodgrass" ;
            ns1:middle "P" ] .
```

JSON to RDF Conversion Steps

1. Convert predicates to URI's
2. Add subject URI's where appropriate
3. Add data types, language tags and/or direction
4. Add list ordering
5. Flesh out object links

URI Objects

JSON-LD Input | Options | Document URL

```
{
  "@id": "http://identifiers.r.us/1173418",
  "http://example.org/person/active": true,
  "http://hl7.org/fhir/Person.name": {
    "http://example.org/person/first": "Seymour",
    "http://example.org/person/middle": "P",
    "http://example.org/person/last": "Snodgrass"
  },
  "http://example.org/person/father": "http://worldfathers.org/id#119341",
  "http://example.org/person/birthdate": {
    "@value": "2002-07-21",
    "@type": "http://www.w3.org/2001/XMLSchema#date"
  },
  "http://example.org/person/children": {
  "@list": [
      "Tom",
      "Dick",
      "Harry"
    ]
  }
}
```

```
"http://example.org/person/father": "http://worldfathers.org/id#119341",
```

Expanded | Compacted | Flattened | Framed | N-Quads | Normalized | Table | Visualized | Signed with RSA | Signed with Bitcoin

```
<http://identifiers.r.us/1173418> <http://example.org/person/active> "true"^^<http://www.w3.org/2001/XMLSchema#boolean> .
<http://identifiers.r.us/1173418> <http://example.org/person/birthdate> "2002-07-21"^^<http://www.w3.org/2001/XMLSchema#date> .
<http://identifiers.r.us/1173418> <http://example.org/person/children> _:b1 .
<http://identifiers.r.us/1173418> <http://example.org/person/father> "http://worldfathers.org/id#119341" .
<http://identifiers.r.us/1173418> <http://hl7.org/fhir/Person.name> _:b0 .
_:b0 <http://example.org/person/first> "Seymour" .
_:b0 <http://example.org/person/last> "Snodgrass" .
_:b0 <http://example.org/person/middle> "P" .
_:b1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> "Tom" .
_:b1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:b2 .
_:b2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> "Dick" .
_:b2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:b3 .
_:b3 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> "Harry" .
_:b3 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> <http://www.w3.org/1999/02/22-rdf-syntax-ns#nil> .
```

https://tinyurl.com/smlfzpt

We've now converted our JSON to RDF

Expanded | Compacted | Flattened | Framed | N-Quads


```
[
  {
    "@id": "http://identifiers.r.us/1173418",
    "http://example.org/person/active": [
      {
        "@value": true
      }
    ],
    "http://example.org/person/birthdate": [
      {
        "@type": "http://www.w3.org/2001/XMLSchema#date",
        "@value": "2002-07-21"
      }
    ],
    "http://example.org/person/children": [
      {
        "@list": [
          {
            "@value": "Tom"
          },
          {
            "@value": "Dick"
          },
          {
            "@value": "Harry"
          }
        ]
      }
    ],
    "http://example.org/person/father": [
      {
        "@value": "http://worldfathers.org/id#119341"
      }
    ],
    "http://hl7.org/fhir/Person.name": [
      {
        "http://example.org/person/first": [
          {
            "@value": "Seymour"
          }
        ],
        "http://example.org/person/last": [
          {
            "@value": "Snodgrass"
          }
        ],
        "http://example.org/person/middle": [
          {
            "@value": "P"
          }
        ]
      }
    ]
  }
]
```


Uhhh…. and this works how?

Remember
Just another RDF Serialization Format…?

JSON-LD Context

The “secret sauce”

Plain ‘ol JSON

```
{
  “name": “BigCocoInc",
  "type": "Company",
  "people": {
     "Sam": {
        "name": {
         first": "Sam",
        "last": "Smith"
      },
     "employees": ["Melissa", "Dazhi"]
   },
  "Melissa": {
     "name": {
       "last": "Johnson"
     }
   },
  "Dazhi": {
  }
}
```

Context

```
{
  "@context": {
    "sdo": "http://schema.org/",
    "foaf": "http://xmlns.com/foaf/0.1/",
    "co": "http://companies.com/",
    "@base": "http://companies.com",
    "type": "@type",
    "name": "@id",
    "people": {
      "@id": "sdo:employee",
      "@container": "@id",
      "@context": {
           "name": "sdo:name",
        "first": "foaf:givenName",
        "last": "foaf:familyName",
        "employees": {
          "@reverse": "co:reports_to",
          "@type": "@id"
        }
      }
    }
```

RDF

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix co: <http://companies.com/> .
@prefix sdo: <http://schema.org/> .

co:BigCocoInc a co:Company ;
    sdo:employee co:Dazhi,
        co:Melissa,
        co:Sam .

co:Dazhi co:reports_to co:Sam .

co:Melissa co:reports_to co:Sam ;
    sdo:name [ foaf:familyName "Johnson" ] .

co:Sam sdo:name [ foaf:familyName "Smith" ;
            foaf:givenName "Sam" ] .
```

http://tinyurl.com/tbmkhzp

@context

The “secret sauce”

@context

A set of declarative rules that tell us how to do everything we did earlier and some more

- define the default vocabulary

Predicates to URIs
We'll start with a default …

JSON-LD Input | Options

```
{
  "@context": {
    "@vocab": "http://example.c
  },
  "id": "1173418",
  "active": true,
  "name": {
    "first": "Seymour",
    "middle": "P",
    "last": "Snodgrass"
  },
  "father": "119341",
  "birthdate": "2002-07-21",
  "children": [
    "Tom",
    "Dick",
    "Harry"
  ]
}
```

```
"@context": {
  "@vocab": "http://example.org/person/id/"
},
"id": "1173418"
```

```
{
  "http://example.org/person/id/active": true,
  "http://example.org/person/id/birthdate": "2002-07-21",
  "http://example.org/person/id/children": [
    "Tom",
    "Dick",
    "Harry"
  ],
  "http://example.org/person/id/father": "119341",
  "http://example.org/person/id/id": "1173418",
  "http://example.org/person/id/name": {
    "http://example.org/person/id/first": "Seymour",
    "http://example.org/person/id/last": "Snodgrass",
    "http://example.org/person/id/middle": "P"
  }
}
```

Get our namespaces under control

```
{
  "@context" : {
    "skos": "http://snomed.info/id/",
    "rdf": "http://snomed.info/id/",
    "xsd": "http://www.w3.org/2001/XMLSchema",
    "ex": "http://example.org/person/",
    "idsrus": "http://identifiers.r.us/"
  }
}
```

Added to github:

https://raw.githubusercontent.com/fhircat/fhir_rdf_validator/master/tutorial/us2ts/example2_namespaces.context.json

Note: nested @contexts (@contexts that reference @contexts) do not work in github due to CORS issues… and tinyurl's as well :-(

Restated w/ Namespaces

JSON-LD Input | Options | Docum

```
{
  "@context": [
    "https://raw.githubusercontent.com/fhircat
/fhir_rdf_validator/master/tutorial/us2ts
/example2_namespaces.context.json",
    {
      "@vocab": "ex",
      "id": "@id",
      "@base": "http://identifiers.r.us/",
      "father": {
        "@id": "idsrus:SeymouresName"
      }
    }
  ],
  "id": "1173418",
  "active": true,
  "name": {
    "first": "Seymour",
    "middle": "P",
    "last": "Snodgrass"
  },
  "father": "119341",
  "birthdate": "2002-07-21",
  "children": [
    "Tom",
    "Dick",
    "Harry"
  ]
}
```

```
{
  "http://example.org/person/id/active": true,
  "http://example.org/person/id/birthdate": "2002-07-21",
  "http://example.org/person/id/children": [
    "Tom",
    "Dick",
    "Harry"
  ],
  "http://example.org/person/id/father": "119341",
  "http://example.org/person/id/id": "1173418",
  "http://example.org/person/id/name": {
    "http://example.org/person/id/first": "Seymour",
    "http://example.org/person/id/last": "Snodgrass",
    "http://example.org/person/id/middle": "P"
  }
}
```

https://tinyurl.com/wc3959o

Non-default maps

Examples: Person | Event | Place | Product | Recipe | Library

JSON-LD Input | Options

```
{
  "@context": [
    "https://raw.githubusercontent.com/fhircat/fhir_rdf_validator/ma
    {
      "@vocab": "ex",
      "id": "@id",
      "@base": "http://identifiers.r.us/",
      "name": {
        "@id": "http://schema.org/name",
        "@context": {
          "first": "http://xmlns.com/foaf/0.1/givenName",
          "middle": "http://schema.org/additionalName",
          "last": "http://xmlns.com/foaf/0.1/familyName"
        }
      },
      "father": "http://purl.obolibrary.org/obo/RoleO_000000",
      "birthdate": "http://hl7.org/fhir/Patient.birthdate",
      "children": "http://purl.obolibrary.org/obo/RoleO_0000008"

    }
  ],
  "id": "1173418",
  "active": true,
  "name": {
    "first": "Seymour",
    "middle": "P",
    "last": "Snodgrass"
  },
  "father": "119341",
  "birthdate": "2002-07-21",
  "children": [
    "Tom",
    "Dick",
    "Harry"
  ]
}
```

https://tinyurl.com/tldozto

Expanded | Compacted | Flattened | Framed | N-Quads | Normalized | Table | Visual

```
<http://identifiers.r.us/1173418> <http://example.org/person/active> "true"^^<http://www.w3.o
<http://identifiers.r.us/1173418> <http://hl7.org/fhir/Patient.birthdate> "2002-07-21" .
<http://identifiers.r.us/1173418> <http://purl.obolibrary.org/obo/RoleO_0000008> "Dick" .
<http://identifiers.r.us/1173418> <http://purl.obolibrary.org/obo/RoleO_0000008> "Harry" .
<http://identifiers.r.us/1173418> <http://purl.obolibrary.org/obo/RoleO_0000008> "Tom" .
<http://identifiers.r.us/1173418> <http://purl.obolibrary.org/obo/RoleO_000000> "119341" .
<http://identifiers.r.us/1173418> <http://schema.org/name> _:c14n0 .
_:c14n0 <http://schema.org/additionalName> "P" .
_:c14n0 <http://xmlns.com/foaf/0.1/familyName> "Snodgrass" .
_:c14n0 <http://xmlns.com/foaf/0.1/givenName> "Seymour" .
```

Add subjects

JSON-LD Input | Options | Document URL

```
{
  "@context": {
    "@vocab": "http://example.org/person/id/"
    "id": "@id",
    "@base": "http://identifiers.r.us/",
    "father": {
      "@id": "http://identifiers.r.us/SeymouresName"
    }
  },
  "id": "1173418",
  "active": true,
  "name": {
    "first": "Seymour",
    "middle": "P",
    "last": "Snodgrass"
  },
  "father": "119341",
  "birthdate": "2002-07-21",
  "children": [
    "Tom",
    "Dick",
    "Harry"
  ]
}
```

```
"id": "@id",
"@base": "http://identifiers.r.us/",
```

```
"@id": "http://id.r.us/SeymouresName"
```

Expanded | Compacted | Flattened | Framed | N-C

```
{
  "@id": "http://identifiers.r.us/1173418",
  "http://example.org/person/id/active": true,
  "http://example.org/person/id/birthdate": "2002-07-21"
  "http://example.org/person/id/children": [
    "Tom",
    "Dick",
    "Harry"
  ],
  "http://example.org/person/id/name": {
    "http://example.org/person/id/first": "Seymour",
    "http://example.org/person/id/last": "Snodgrass",
    "http://example.org/person/id/middle": "P"
  },
  "http://identifiers.r.us/SeymouresName": "119341"
}
```

Adding Datatypes

JSON-LD Input | Options

```
{ "@context": {
    "@vocab": "http://ex.org/builtins/",
    "xsd": "http://www.w3.org/2001/XMLSchema#",
    "strreal": {"@type": "xsd:decimal"},
    "strurl": {"@type": "xsd:anyURI"},
    "strdate": {"@type": "xsd:date"},
    "strdbl": {"@type": "xsd:double"}
  },

  "@id": "http://ex.org/builtins",
  "str": "abc",
  "strreal": 17.2,
  "strurl": "http://google.org/",
  "strdate": "2012-01-17",
  "strint": 1,
  "strnegint": -17,
  "strdbl": 245.999631999631,
  "strbool": false,
  "strnull": null
}
```

```
"@context": {
  "@vocab": "http://ex.org/builtins/",
  "xsd": "http://www.w3.org/2001/XMLSchema#",
  "strreal": {"@type": "xsd:decimal"},
  "strurl": {"@type": "xsd:anyURI"},
  "strdate": {"@type": "xsd:date"},
  "strdbl": {"@type": "xsd:double"}
},
```

Expanded | Compacted | Flattened | Framed | N-Quads | Normalized | Table | Visualized | Signed with RSA | Sign

```
<http://ex.org/builtins> <http://ex.org/builtins/str> "abc" .
<http://ex.org/builtins> <http://ex.org/builtins/strbool> "false"^^<http://www.w3.org/2001/XMLSchema#boolean> .
<http://ex.org/builtins> <http://ex.org/builtins/strdate> "2012-01-17"^^<http://www.w3.org/2001/XMLSchema#date> .
<http://ex.org/builtins> <http://ex.org/builtins/strdbl> "2.45999631999631E2"^^<http://www.w3.org/2001/XMLSchema#double> .
<http://ex.org/builtins> <http://ex.org/builtins/strint> "1"^^<http://www.w3.org/2001/XMLSchema#integer> .
<http://ex.org/builtins> <http://ex.org/builtins/strnegint> "-17"^^<http://www.w3.org/2001/XMLSchema#integer> .
<http://ex.org/builtins> <http://ex.org/builtins/strreal> "1.72E1"^^<http://www.w3.org/2001/XMLSchema#decimal> .
<http://ex.org/builtins> <http://ex.org/builtins/strurl> "http://google.org/"^^<http://www.w3.org/2001/XMLSchema#anyURI> .
```

Adding Datatypes

JSON-LD Input | Options | Docur

```
{
  "@context": [
    "https://raw.githubusercontent.com/fhircat
/fhir_rdf_validator/master/tutorial/us2ts
/example2_namespaces.context.json",
    {
      "@vocab": "ex",
      "id": "@id",
      "@base": "http://identifiers.r.us/",
      "father": {
        "@id": "idsrus:SeymouresName"
      },
      "birthdate": {
        "@type": "xsd:date"
      }
    }
  ],
  "id": "1173418"
```

```
  "birthdate": {
    "@type": "xsd:date"
  }
```

Expanded | Compacted | Flattened | Framed | N-

```
{
  "@id": "http://identifiers.r.us/1173418",
  "http://example.org/person/active": true,
  "http://example.org/person/birthdate": {
    "@type": "http://www.w3.org/2001/XMLSchema#date",
    "@value": "2002-07-21"
  },
  "http://example.org/person/children": [
    "Tom",
    "Dick",
    "Harry"
  ],
  "http://example.org/person/name": {
    "http://example.org/person/first": "Seymour",
    "http://example.org/person/last": "Snodgrass",
    "http://example.org/person/middle": "P"
  },
  "http://identifiers.r.us/SeymouresName": "119341"
}
```

Languages
Option 1

JS


```
{
  "@context": {
    "id": "@id",
    "sct": "http://snomed.info/id/",
    "sctid": "sct",
    "skos": "http://www.w3.org/2004/02/skos/core#",
    "@base": "http://snomed.info/id/",
    "fsn_en": {"@id": "sct::9000000000000003001", "@language": "en"},
    "fsn_fr_ca": {"@id": "sct::9000000000000003001", "@language": "fr-ca"},
    "pref_en": {"@id": "skos:prefLabel", "@language": "en"},
    "pref_da": {"@id": "skos:prefLabel", "@language": "da"},
    "alt_en_1": {"@id": "skos:altLabel", "@language": "en"},
    "alt_en_2": {"@id": "skos:altLabel", "@language": "en"},
    "fsn_fr_ca": {"@id": "skos:altLabel", "@language": "fr-ca"}
  },
  "id": "14189004",
  "fsn_en": "Measles (disorder)",
  "pref_en": "Measles",
  "alt_en_1": "Morbilli",
  "alt_en_2": "Rubeola",
  "fsn_fr_ca": "rugeole (trouble)",
  "alt_fr_ca": "rugeole",
```


https://tinyurl.com/wbuu8rk

Expanded | Compacted | Flattened | Framed | N-Quads | Normalized | Table | Visualized | Signed with R


```
<http://snomed.info/id/14189004> <http://snomed.info/id/:9000000000000003001> "Measles (disorder)"@en .
<http://snomed.info/id/14189004> <http://www.w3.org/2004/02/skos/core#altLabel> "Morbilli"@en .
<http://snomed.info/id/14189004> <http://www.w3.org/2004/02/skos/core#altLabel> "Rubeola"@en .
<http://snomed.info/id/14189004> <http://www.w3.org/2004/02/skos/core#altLabel> "rugeole (trouble)"@fr-ca .
<http://snomed.info/id/14189004> <http://www.w3.org/2004/02/skos/core#prefLabel> "Measles"@en .
<http://snomed.info/id/14189004> <http://www.w3.org/2004/02/skos/core#prefLabel> "Mæslinger"@da .
```

Languages
Option 2

JSON-LD Input | Options

```
{
  "@context": {
    "id": "@id",
    "sct": "http://snomed.info/id/",
    "sctid": "sct",
    "skos": "http://www.w3.org/2004/02/skos/core#",
    "@base": "http://snomed.info/id/",

    "descriptions": "@nest",
    "fsn": {"@id": "sct:90000000000003001", "@container": "@language"},
    "pref": {"@id": "skos:prefLabel", "@container": "@language"},
    "alt": {"@id": "skos:altLabel", "@container": "@language"}
  },
  "id": "14189004",
  "descriptions": {
    "fsn": {
       "en": "Measles (disorder)",
       "fr-ca": "rugeole (trouble)"
    },
    "pref": {
       "en": "Measles",
      " da": "Mæslinge"
    },
    "alt": {
      "en": [
          "Morbilli",
          "Rubeola"
```

"@nest"

"@container": "@language"

https://tinyurl.com/w63bj5z

```
<http://snomed.info/id/14189004> <http://snomed.info/id/90000000000003001> "Measles (disorder)"@en .
<http://snomed.info/id/14189004> <http://snomed.info/id/90000000000003001> "rugeole (trouble)"@fr-ca .
<http://snomed.info/id/14189004> <http://www.w3.org/2004/02/skos/core#altLabel> "Morbilli"@en .
<http://snomed.info/id/14189004> <http://www.w3.org/2004/02/skos/core#altLabel> "Rubeola"@en .
<http://snomed.info/id/14189004> <http://www.w3.org/2004/02/skos/core#altLabel> "rugeole"@fr-ca .
<http://snomed.info/id/14189004> <http://www.w3.org/2004/02/skos/core#prefLabel> "Measles"@en .
<http://snomed.info/id/14189004> <http://www.w3.org/2004/02/skos/core#prefLabel> "Mæslinge"@ da .
```

List Ordering

JSON-LD Input | Options

```
{
  "@context": [
    "https://raw.githubusercontent.com/fhircat/
    {
        "@vocab": "ex",
        "id": "@id",
        "@base": "http://identifiers.r.us/",
        "father": {
          "@id": "idsrus:SeymouresName"
        },
        "children": {
          "@container": "@list"
        }
      }
    ],
    "id": "1173418",
    "active": true,
    "name": {
      "first": "Seymour",
      "middle": "P",
      "last": "Snodgrass"
    },
    "father": "119341",
    "birthdate": "2002-07-21",
    "children": [
      "Tom",
      "Dick",
      "Harry"
    ]
}
```

```
    },
    "children": {
      "@container": "@list"
    }
}
```

```
{
  "@id": "http://identifiers.r.us/1173418",
  "http://example.org/person/active": true,
  "http://example.org/person/birthdate": "2002-07-21",
  "http://example.org/person/children": {
    "@list": [
      "Tom",
      "Dick",
      "Harry"
    ]
  },
  "http://example.org/person/name": {
    "http://example.org/person/first": "Seymour",
    "http://example.org/person/last": "Snodgrass",
    "http://example.org/person/middle": "P"
  },
  "http://identifiers.r.us/SeymouresName": "119341"
}
```

https://tinyurl.com/ugscfur

URI Objects

```
{
  "@context": [
    "https://raw.githubusercontent.com/fhircat/fhir_rdf_validator/m
    {
        "@vocab": "ex",
        "id": "@id",
        "@base": "http://identifiers.r.us/",
        "father": {
          "@type": "@id",
          "@context": {
              "@base": "http://worldfathers.org/i
          }
        },
        "children": {
          "@container": "@list"
        }
      }
    ],
    "id": "1173418
    "active": true
    "name": {
      "first": "Se
      "middle": "P
      "last": "Sno
    },
    "father": "119341",
    "birthdate": "2002-07-21",
    "children": [
      "Tom",
      "Dick",
      "Harry"
    ]
}
```

```
"father": {
  "@type": "@id",
  "@context": {
      "@base": "http://worldfathers.org/id#"
  }
}
```

Expanded | Compacted | Flattened | Framed | N-Quads | Normalized | Table | Visualized | Sig

```
<http://identifiers.r.us/1173418> <http://example.org/person/active> "true"^^<http://www.w3.org/2001/XM
<http://identifiers.r.us/1173418> <http://example.org/person/birthdate> "2002-07-21" .
<http://identifiers.r.us/1173418> <http://example.org/person/children> _:b1 .
<http://identifiers.r.us/1173418> <http://example.org/person/father> <http://worldfathers.org/119341> .
<http://identifiers.r.us/1173418> <http://example.org/person/name> _:b0 .
_:b0 <http://example.org/person/first> "Seymour" .
_:b0 <http://example.org/person/last> "Snodgrass" .
```

https://tinyurl.com/t8bdl39

So, where are we?

We have our *original* JSON

JSON-LD Input | Options


```
{
  "id": "1173418",
  "active": true,
  "name": {
    "first": "Seymour",
    "middle": "P",
    "last": "Snodgrass"
  },
  "father": "119341",
  "birthdate": "2002-07-21",
  "children": [
    "Tom",
    "Dick",
    "Harry"
  ]
}
```


https://tinyurl.com/vvqd8yg

We add a context link

JSON-LD Input | Options | https://tinyurl.com/wkunpws | Docum

```
{
  "@context": [
    "https://raw.githubusercontent.com/fhircat/fhir_rdf_validator/master/tutorial/us2ts/example2_namespaces.context.json",
    "https://raw.githubusercontent.com/fhircat/fhir_rdf_validator/master/tutorial/us2ts/person.context.json"
  ],
  "id": "1173418",
  "active": true,
  "name": {
    "first": "Seymour",
    "middle": "P",
    "last": "Snodgrass"
  },
  "father": "119341",
  "birthdate": "2002-07-21",
  "children": [
    "Tom",
    "Dick",
    "Harry"
  ]
}
```

Note 1: The reason this is *two* links is strictly the CORS issue

Note 2: There are several ways to add the context without touching the JSON (!)

And we have RDF!!!

Expanded | Compacted | Flattened | Framed | N-Quads | Normalized | Table | Visualized | Signed with RSA | Signed with Bitco

```
<http://identifiers.r.us/1173418> <http://example.org/person/active> "true"^^<http://www.w3.org/2001/XMLSchema#boolean> .
<http://identifiers.r.us/1173418> <http://example.org/person/birthdate> "2002-07-21"^^<http://www.w3.org/2001/XMLSchema#date> .
<http://identifiers.r.us/1173418> <http://example.org/person/children> _:b1 .
<http://identifiers.r.us/1173418> <http://example.org/person/father> <http://worldfathers.org/119341> .
<http://identifiers.r.us/1173418> <http://example.org/person/name> _:b0 .
_:b0 <http://example.org/person/first> "Seymour" .
_:b0 <http://example.org/person/last> "Snodgrass" .
_:b0 <http://example.org/person/middle> "P" .
_:b1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> "Tom" .
_:b1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:b2 .
_:b2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> "Dick" .
_:b2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> _:b3 .
_:b3 <http://www.w3.org/1999/02/22-rdf-syntax-ns#first> "Harry" .
_:b3 <http://www.w3.org/1999/02/22-rdf-syntax-ns#rest> <http://www.w3.org/1999/02/22-rdf-syntax-ns#nil> .
```

Expanded | Compacted | Flattened | Framed | N-Quads

```
[
  {
    "http://example.org/person/active": [
      {
        "@value": true
      }
    ],
    "http://example.org/person/birthdate": [
      {
        "@type": "http://www.w3.org/2001/XMLSchema#date",
        "@value": "2002-07-21"
      }
    ],
    "http://example.org/person/children": [
      {
        "@list": [
          {
            "@value": "Tom"
          },
          {
            "@value": "Dick"
          },
          {
            "@value": "Harry"
          }
        ]
      }
    ],
    "http://example.org/person/father": [
      {
        "@id": "http://worldfathers.org/119341"
      }
    ],
    "@id": "http://identifiers.r.us/1173418",
    "http://example.org/person/name": [
      {
        "http://example.org/person/first": [
          {
            "@value": "Seymour"
          }
        ],
        "http://example.org/person/last": [
          {
            "@value": "Snodgrass"
          }
```

```
@prefix ns1: <http://example.org/person/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://identifiers.r.us/1173418> ns1:active true ;
    ns1:birthdate "2002-07-21"^^xsd:date ;
    ns1:children ( "Tom" "Dick" "Harry" ) ;
    ns1:father <http://worldfathers.org/119341> ;
    ns1:name [ ns1:first "Seymour" ;
               ns1:last "Snodgrass" ;
               ns1:middle "P" ] .
```

```
yntax-ns#"

99e2fae4326a8f74b3">
ae4326a8f74b4"/>

99e2fae4326a8f74b4">
  <rdf:first>Harry</rdf:first>
  <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
</rdf:Description>
<rdf:Description rdf:about="http://identifiers.r.us/1173418">
  <ns1:active rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">true</ns1:active>
  <ns1:name rdf:nodeID="fc14735b5214c498699e2fae4326a8f74b2"/>
  <ns1:father rdf:resource="http://worldfathers.org/119341"/>
  <ns1:birthdate rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2002-07-21</ns1:birthdate>
  <ns1:children rdf:nodeID="fc14735b5214c498699e2fae4326a8f74b1"/>
</rdf:Description>
<rdf:Description rdf:nodeID="fc14735b5214c498699e2fae4326a8f74b2">
```

But wait! There's more

JSON-LD Framing

Makes it bidirectional (!)

Plain ‘ol JSON

```
{
 ,
  "name": "BigCocoInc",
  "type": "Company",
  "people": {
   "Sam": {
    "name": {
     "first": "Sam",
     "last": "Smith"
    },
    "employees": [
     "Melissa",
     "Dazhi"
    ]
   },
   "Melissa": {
    "name": {
     "last": "Johnson"
    }
   },
   "Dazhi": {}
 }
}
```

http://tinyurl.com/tgxu78k

RDF

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix co: <http://companies.com/> .
@prefix sdo: <http://schema.org/> .

co:BigCocoInc a co:Company ;
    sdo:employee co:Dazhi,
        co:Melissa,
        co:Sam .

co:Dazhi co:reports_to co:Sam .

co:Melissa co:reports_to co:Sam ;
    sdo:name [ foaf:familyName "Johnson" ] .

co:Sam sdo:name [ foaf:familyName "Smith" ;
            foaf:givenName "Sam" ] .
```

Frame

```
{ "@context": [
  "https://raw.githubusercontent.com/fhircat/fhir_rdf_validator/master/tutorial/
company.context.jsonld",
   {
    "@vocab": "http://company.com/",
    "@base": "http://company.com/"
   }
 ],
 "@type": "co:Company"
}
```

Summary

- JSON-LD expanded syntax is just another RDF serialization
- JSON-LD *context* is a map between JSON and JSON-LD expanded syntax
- Contexts can be completely separate from the JSON itself — developers don’t even have to know they exist!
    - Contexts can be viewed as a *semantics map*(!)
- Mapping between JSON and RDF is *bidirectional*
    - *Y*ou can start with JSON and emit (context specific!) RDF
    - You can start with triples from SPARQL, RDFa, … and create JSON(!)

Links and the Like

- https://www.w3.org/TR/rdf11-primer/ — the RDF primer
- https://www.w3.org/TR/xmlschema11-2/ — XML Schema Data Types
- https://www.json.org/json-en.html — Everything JSON
- http://json-ld.org — the on-ramp to everything JSON-LD
- https://json-ld.org/playground/ — the wonderful playground
- https://github.com/digitalbazaar/jsonld.js — the playground source
- https://github.com/fhircat/fhir_rdf_validator/tree/master/tutorial/us2ts — temporary holding spot for tutorial materials

Credits

This work is supported by funding from:

- NIH FHIRCat (R56 EB028101)

- CG Chute, MA Haendel, CJ Mungall. TransMed: A translational data integration machine for biomedical discovery. NIH/NCATS 1OT3TR002019, in response to Biomedical Data Translator: Technical Feasibility Assessment and Architecture Design Projects (OT3) OT-TR-16-001, 2016.